 ISO/IEC JTC 1/SC 29/WG 7 N514

**ISO/IEC JTC 1/SC 29/WG 7  
MPEG 3D Graphics Coding   
Convenorship: AFNOR (France)**

**Document type:** Output Document

**Title:** Preliminary Dataset Collection for AI-based Point Cloud Experiments

**Status:** Approved

**Date of document:** 2023-02-10

**Source:** ISO/IEC JTC 1/SC 29/WG 7

**Expected action:** None

**Action due date:** None

**No. of pages:** 12 (with cover page)

**Email of Convenor:** marius.preda @ imt . fr

**Committee URL:** [https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg7](https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3)

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 7 MPEG 3D Graphics Coding**

**ISO/IEC JTC 1/SC 29/WG 7 N514**

**Jan 2023, Virtual**

|  |  |
| --- | --- |
| **Title** | **Preliminary Dataset Collection for AI-based Point Cloud Experiments** |
| **Source** | **WG7, MPEG 3D Graphics Coding** |
| **Status** | **Approved** |
| **Serial Number** | **w22431** |

Abstract

An exploration AhG on “AI-based coding for graphics” was established during the 4th WG7 /the 135th MPEG) meeting in July 2021. This document provides a description of the available open-source point cloud datasets and their applicable terms of use.

# Introduction

One of the most important aspects of learning-based systems is the dataset that is used by a learning-based method to learn the task. In this document we discuss some existing open-source datasets that can be considered within the MPEG group for the purpose of learning-based point cloud compression and analysis.

# Proposed Datasets terms of use

We propose the following four datasets from the plethora of existing open-source datasets: ShapeNetCore dataset, ModelNet40 dataset, SemanticKitti dataset, and Ford sequences from GPCC CTC. The first two datasets, ShapeNetCore and ModelNet, are CAD models of individual objects whereas Kitti and Ford are LiDAR scans of outdoor scenes. As highlighted in the MPEG input document [1], these datasets can be put in the following categories based on the type of data and point density: *Surface point clouds* for ModelNet40 and ShapeNetCore; and *Automotive Frame point clouds* for Kitti and Ford.

In addition to these datasets, for dynamic point cloud coding the 8i, Technicolor, Owlii datasets that are part of the V-PCC CTC [2] can be used.

We now provide some usage conditions specific to each dataset below. Ford, 8i, Technicolor, and Owlii sequences are already part of MPEG CTC and are thus skipped.

## ShapeNet

ShapeNet terms of use can be found at [3] :

* Two out of the seven terms and conditions are highlighted below:
  + Researcher shall use the Database only for non-commercial research and educational purposes.
  + If Researcher is employed by a for-profit, commercial entity, Researcher's employer shall also be bound by these terms and conditions, and Researcher hereby represents that he or she is fully authorized to enter into this agreement on behalf of such employer.
* The group needs to discuss if these usage conditions allow usage of this data for training and evaluation in regard to learning-based methods.

## ModelNet

ModelNet copyright can be found at [4] and is reproduced below:

* “All CAD models are downloaded from the Internet and the original authors hold the copyright of the CAD models. The label of the data was obtained by us via Amazon Mechanical Turk service and it is provided freely. This dataset is provided for the convenience of academic research only.”
* The group needs to evaluate if this copyright allows usage of this data for training and evaluation in regard to learning-based methods.

## Kitti, SemanticKitti

Kitti dataset copyright can be found in [5]:

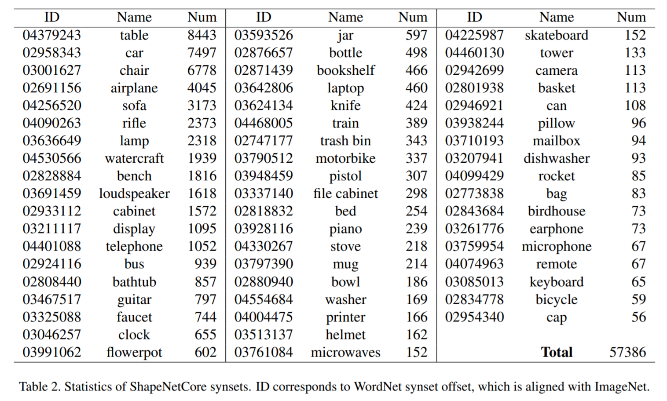
* “All datasets and benchmarks on this page are copyright by us and published under the [Creative Commons Attribution-NonCommercial-ShareAlike 3.0](http://creativecommons.org/licenses/by-nc-sa/3.0/) License. This means that you must attribute the work in the manner specified by the authors, you may not use this work for commercial purposes and if you alter, transform, or build upon this work, you may distribute the resulting work only under the same license.”
* The group needs to evaluate if this license allows usage of this data for training and evaluation in regard to learning-based methods.

# Surface point cloud dataset characteristics

Surface point cloud dataset composed of single object can be used for training networks designed to: a) compress this object level data, and b) perform vision tasks such as object recognition, object segmentation, object part segmentation, etc.

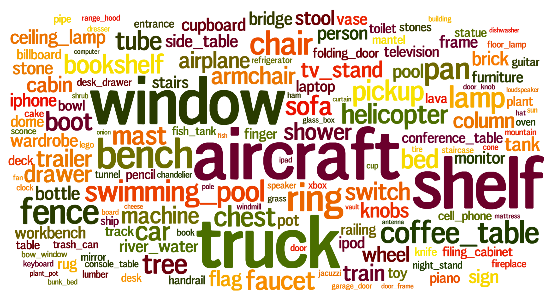
## ShapeNetCore Dataset

ShapeNet is an ongoing effort to establish a richly-annotated, large-scale dataset of 3D shapes. ShapeNetCore dataset, a subset of the full ShapeNet dataset, contains 51,300 3D objects models from 55 common object categories. The data is provided as 3D meshes (dense connected graphs with faces) which can be sampled to obtain point clouds using several tools available on the ShapeNet website. PyTorch3D now also supports easy loading of ShapeNetCore dataset for that purpose. A method for mesh sampling is recommended as provided by “pytorch\_geometric” library (see Appendix 7.1). Example objects in ShapeNetCore are highlighted in the following table taken from [6].



## ModelNet dataset

ModelNet is a large 3D CAD model dataset containing various shapes with 151,128 models in 660 unique object categories. Two smaller subsets, ModelNet10 and ModelNet40 are also available each containing 10 and 40 unique object categories. The CAD models are in the format of “OFF” (object file format) files, which are mesh files from which point clouds can be sampled with required sampling density. A method for mesh sampling is recommended as provided by “pytorch\_geometric” library (see Appendix 7.1). Below is a word cloud representing the various categories in the original ModelNet dataset.



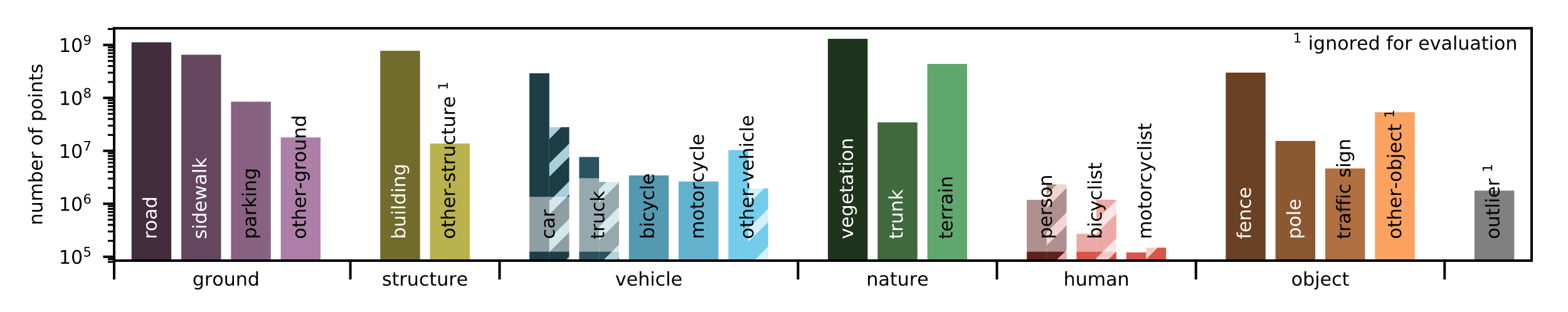
# Automotive Frame point cloud dataset characteristics

Automotive Frame point clouds are LiDAR point clouds obtained from sensors mounted on automotive driving outdoors. These point clouds consist of a sequence of frames obtained in a series with a sensor specified fps (frames per second) parameter. This kind of data can be used to train networks to: a) compress LiDAR data, and b) perform specific machine tasks such as object recognition, object segmentation, motion tracking, motion segmentation, semantic segmentation, etc.

## Kitti Dataset

Kitti dataset is a large-scale dataset with LiDAR scans obtained in outdoor self-driving settings containing large amount of raw LiDAR data, along with other data modalities (e.g., RGB) and some labels. SemanticKitti developed by a group of researchers from University of Bonn, contains dense annotations for all the sequences in the odometry task (from within the original Kitti dataset). Out of 43552 scans in total from Kitti, SemanticKitti provides ground truth annotations under several categories for 23,311 scans for training purposes.

The data itself is provided as “.bin” files in continuous range (-96,96). However, with conversion to integer values the data lies within range (0,2^16). The average number of points in each scan is around ~120K.

Below is an example of the kind of dense (point-wise) annotations available in SemanticKitti.

## Ford sequences from GPCC CTC

Re-use of a refined current MPEG dataset for testing was proposed in [7]. We propose to use the three frame-based “Ford” sequences [8] which contain 18bit data with 1500 frames in each sequence. Moreover, each sequence has on average ~100K points per frame.

# Dataset partitioning

The point clouds data in the datasets should be divided into two categories:

* Test set: This set of data is entirely reserved for benchmarking the performance of the learning-based PCC codec. It should not be used for training.
* Trainval set: This set of data is used to train and validate the learning-based codec. It can be used freely.

## Surface Point Clouds (Category 1: Static Objects and Scenes)

In the 6th WG7/137th MPEG meeting a preliminary dataset split for surface point clouds was discussed. Table 5‑1 and Table 5‑2 list the proposed trainval and test sets derived from the Common Test Conditions (CTC) for G-PCC[[1]](#footnote-2) [8]. Details about the methodology applied to compose the datasets are presented in the document “[AI-3DGC][EE13.54-ralated] Dataset Split for AI-PCC” [9]. The main concepts are summarized in Annex 8.2.

The idea behind the definition of the proposed dataset split is to build a collection of test point clouds with the largest possible variety of geometry bit-depths and sparseness levels. Table 5.3 shows in green, the bit-depths/sparseness subclasses covered by the proposed split of G-PCC’s CTC dataset [8]; in orange are the missing subclasses that are completed with mesh-derived point clouds; and in red are the subclasses that were not addressed.

ModelNet and ShapeNet are reserved exclusively for training and validation, and any pre-processing strategy applied to the trainval set (point cloud or mesh) is not mandatory and may be freely specified by the proponents, as long as the means to reproduce the training procedure for crosschecking purposes is provided. For more details, see document “Guidelines for conducting AI exploration experiments for PCC [10]”.

Table ‑ Trainval set derived from “Common test Conditions for G-PCC.”

Table

Description automatically generated

Table ‑ Test set derived from “Common test Conditions for G-PCC.”

Table

Description automatically generated

## Dynamic Objects (Category 2) and Dynamic Acquisition (Category 3)

A more detailed analysis and dataset split for dynamic point clouds and dynamic acquired point clouds (LiDAR/automotive-frame and Automotive-fused) needs to be performed. For LiDAR automotive-frame point clouds, Kitti and Ford sequences are recommended. The exact data split for Ford and Kitti point clouds is presented below. A study for automotive-fused dataset definition and split is still required.

### Dynamic Point Clouds (Category 2)

At the 6th WG7/138th MPEG meeting the EE5.3 on AI-based dynamic point cloud coding was established [11]. As part of the EE5.3 activities 10-bit datasets for training and testing were selected, which are listed here in Table 5‑3 and Table 5‑4, respectively. The 11-bit Owlii sequences were converted to 10 bit[[2]](#footnote-3). In addition, it is recommended to use the original 11-bit Owlii sequences for testing as well.

Table ‑ Trainval set derived from “Common test Conditions for V-PCC.”

|  |  |  |
| --- | --- | --- |
| Datasets for training (10-bit) | | |
| Training Sequences | Longdress (8i) | 300 frames |
| Loot (8i) | 300 frames |
| Queen (Technicolor) | 250 frames |
| RedAndBlack (8i) | 300 frames |
| Soldier (8i) | 300 frames |

Table ‑ Test set derived from “Common test Conditions for V-PCC.”

|  |  |  |
| --- | --- | --- |
| Datasets (11-bit originals converted to 10-bit) | | |
| Testing Sequences | basketball\_player\_vox11 | frames 1 to 100 |
| Dancer\_vox11 | frames 1 to 100 |
| exercise\_vox11 | frames 1 to 100 |
| model\_vox11 | frames 1 to 100 |
|  |  |

### Dynamic Acquisition (Category 3)

In the 6th WG7/138th MPEG meeting a preliminary dataset split for LiDAR automotive-frame point clouds was discussed. For LiDAR automotive-frame point clouds, Kitti and Ford sequences are recommended. Ford sequences are part of the MPEG CTC and consists of three separates sequences Ford\_01, Ford\_02 and Ford\_03. All three sequences are very similar in terms of bitrate requirements, thus it is proposed to use Ford\_01 for trainval set and keep Ford\_02 and Ford\_03 for test set.

For the case of Kitti dataset, a split for testing and trainval has been adopted in the existing literature based on some annotations provided later as part of SemanticKitti. It is thus suggested to follow this split and use sequences ‘00’ to ‘10’ for trainval set and keep sequences ‘11’ to ‘21’ for test set. Additionally, since Kitti is not part of MPEG CTC, some test conditions need to be defined for this dataset. These conditions are presented below:

* Kitti dataset is very huge with 20241 scans available for testing in sequences ‘11’ to ‘21’ from the official test set. It is suggested to at least report results for sequences ‘11’, ‘12’ and ‘13’ with the remaining test sequences as optional.
* We suggest to use 18bit quantized KITTI for the test set.
  + “quantize\_resolution” function [12] can be used to quantize the data to specified resolution
* Since Kitti dataset does not come with ground truth normals, the ‘quantize\_resolution’ function also provides functionality to calculate normals along with quantization using an open-source python library called “Point Cloud Utils” [13].
  + For normal calculation, it is suggested to use 32 nearest neighbors. This was observed after testing “Point Cloud Utils” to generate normals for the Ford dataset and comparing with the already available normals.

# Reporting template for AI-based experiments

In the 6th WG7/138th MPEG meeting the need for a Python-based reporting template, in addition to the existing Excel template, was discussed. The advantages and disadvantages of both templates are discussed in the input document m59830 [14]. Based on the discussion during and after the meeting, an initial version of the Python template named “mpeg-pcc-ai-report” [15] was developed having the following capabilities:

* Take two CSV files with results from AI experiments have the same input data as the MPEG Excel reporting template
* Generate the RD curves for each input sequence as PNG files
* Generate the overall RD curve as PNG file
* Print the summary for overall bitrate and PSNR
* Print category-wise summary for bitrate and PSNR

Additional updates to the template will be added over time to refine its features and include more functionalities.

# Proposed mandates to be explored

* The biggest question is for the group to evaluate the usage conditions of the datasets proposed in this document to infer if the proposed datasets can be used within the group.
* Should the AI methods be benchmarked on data that is from the same dataset as the trainval data or a different dataset, or both?
  + If a codec performs well on data from a different dataset than one used in trainval, it shows better generalizability.
* How much data should be used for training?
  + Networks with many parameters will require extensive data, while others can do with less data due to smaller number of parameters.
* Should one model be trained for all rate points or should separate models for each target rate point be trained?
* When evaluating various existing point cloud compression architectures, in [16], the authors decided to compress down-sampled point clouds instead of full resolution point clouds. This should still be a topic of discussion in the next meeting as compression of full resolution point clouds is an important issue.
* We also reiterate the already presented call within the group to gather newer and diverse datasets for training and evaluation of the AI methods.
* Common dataset definition and split for Category 2: Dynamic Objects.
* Common dataset definition and split for Category 3, automotive-fused point clouds.

# Appendix

## Sampling point clouds from mesh models

It is recommend using the sampling method provided by the PyTorch-based library “pytorch\_geometric” under the MIT license. This library is widely used for PyTorch-based learning architectures for 3D datasets. The aim is to uniformly sample points on mesh faces according to their face area. It follows the following steps to sample points on a mesh object:

1. Normalize the positions of the vertices and then calculate area of each face.
2. Normalize the area by the sum of all areas to make a probability density function (pdf) which is used to obtain the number of samples on each face, given the total number of points to be sampled from the mesh altogether.
3. Determine the sampled points from each face by starting from one vertex of the face and adding random vectors to it staying within the face area.
4. Determine the final sampled point positions by denormalizing to account for the initial normalization.

More details and the code for this mesh sampling can be found [online](https://pytorch-geometric.readthedocs.io/en/latest/_modules/torch_geometric/transforms/sample_points.html) within the pytorch\_geometric documentation for “sample\_points.py” file available in the “transforms” folder.

## Local and Global Density, Spread and Homogeneity

This section briefly describes the parameters used to define the proposed dataset split. Details are presented in the document [9].

### Local Density

The local density DL is defined as a function of the number of points that surrounds each point inside a spheric volume. It is calculated by counting for each point the number of neighbors N inside a sphere of radius R, and then dividing N by the neighborhood volume. That is, DL= N / [(4/3).Pi.R3)]. Points with no neighbors have an invalid (*NaN*) local density. A default radius for each point cloud can be computed using the GetPointCloudRadius() method of CloudComPy[[3]](#footnote-4), a Python module to interface to the CloudCompare[[4]](#footnote-5) software. The computeLocalDensity() from the same Python module can be used to compute the local density associated with each point. The default radius and the local density map can also be obtained using the CoudCompare software directly. In this case, visualization tools are available. As an example, Figure 8‑1 shows local density maps, histograms and boxplots for a 10-bit mesh-derived point cloud sampled with different number of points.

|  |  |  |
| --- | --- | --- |
| (a) 4,770 points | (b) 56,432 points | (c) 354,837 points |
| (d) (Spread = 24.2%) | (e) (Spread = 18.3%) | (f) (Spread = 14.2%) |
| (g) | (h) | (i) |
|  | | |

Figure ‑ Density maps, histograms, and boxplots for a 10-bit mesh-derived point cloud.

### Global Density

A histogram can be computed from the local densities of each point, as well as the median and the interquartile range (IQR). The median can be used as an estimate for the global density DGM.

### Spread and Homogeneity

The local density statistics can be further used to determine the degree of homogeneity of a point cloud. Roughly speaking, homogeneous point clouds are those that present a “low” local-density variation. In contrast, heterogeneous point clouds are those that present a “high” local-density variation. One may estimate the interquartile range and use this value to calculate a “spread” metric S in order to determine the degree of homogeneity of a point cloud. The “spread“ is given by S = 100\*IQR/TotalRange %, where TotalRange is the total histogram range. For classification purposes, one can define a threshold T and determine if a point cloud is homogeneous (S <= T) or heterogeneous (S > T).

The distribution of point clouds in subclasses according to bit-depth/sparseness/ homogeneity considering a specific radius per point cloud is shown in the table below.

Table ‑. Classification of point clouds into subcategories of bit-depth/sparseness/homogeneity: spread threshold for homogeneity vs. heterogeneity classification is 11.6%

|  |  |  |  |
| --- | --- | --- | --- |
| bit-depth | Sparseness | Homogeneity | Point Cloud |
| 10 | Solid | homogeneous | queen\_0200  redandblack\_vox10\_1550  loot\_vox10\_1200  longdress\_vox10\_1300  soldier\_vox10\_0690 |
| heterogeneous |  |
| Dense |  | |
| Sparse |
| Scant |
| 11 | Solid | homogeneous | facade\_00064\_vox11  basketball\_player\_vox11\_00000200  dancer\_vox11\_00000001 |
| heterogeneous |  |
| Dense |  | |
| Sparse |
| Scant |
| 12 | Solid | homogeneous | thaidancer\_viewdep\_vox12 |
| heterogeneous |  |
| Dense | homogeneous | soldier\_viewdep\_vox12  loot\_viewdep\_vox12  redandblack\_viewdep\_vox12  longdress\_viewdep\_vox12  boxer\_viewdep\_vox12 |
| heterogeneous | facade\_00009\_vox12  head\_00039\_vox12  frog\_00067\_vox12  house\_without\_roof\_00057\_vox12 |
| Sparse | homogeneous | arco\_valentino\_dense\_vox12  staue\_klimt\_vox12 |
| heterogeneous | shiva\_00035\_vox12  egyptian\_mask\_vox12 |
| Scant |  | |
| 13 | Solid |  | |
| Dense |
| Sparse | homogeneous |  |
| heterogeneous | ulb\_unicorn\_vox13 |
| Scant |  | |
| 14 | Solid |  | |
| Dense | homogeneous | landscape\_00014\_vox14  facade\_00064\_vox14 |
| heterogeneous |  |
| Sparse | homogeneous |  |
| heterogeneous | palazzo\_carignano\_dense\_vox14  facade\_00015\_vox14 |
| Scant |  | |
| 15 | Solid |  | |
| Dense |
| Sparse | homogeneous | ulb\_unicorn\_hires\_vox15 |
| heterogeneous |  |
| Scant |  | |
| 16 | Solid |  | |
| Dense |
| Sparse | homogeneous | stanford\_area\_2\_vox16  stanford\_area\_4\_vox16 |
| heterogeneous |  |
| Scant |  | |
| 20 | Solid |  | |
| Dense |
| Sparse |
| Scant | homogeneous | facade\_00064\_vox20  arco\_valentino\_dense\_vox20  stanford\_area\_2\_vox20  landscape\_00014\_vox20  ulb\_unicorn\_vox20  ulb\_unicorn\_hires\_vox20  stanford\_area\_4\_vox20  staue\_klimt\_vox20  shiva\_00035\_vox20 |
| Heterogeneous | facade\_00009\_vox20  head\_00039\_vox20  palazzo\_carignano\_dense\_vox20  egyptian\_mask\_vox20  frog\_00067\_vox20  facade\_00015\_vox20  house\_without\_roof\_00057\_vox20 |

# References

|  |  |
| --- | --- |
| [1] | K. M. D. Flynn, "G-PCC: Updating common test conditions," *MPEG,* no. m57468, Jul 2021. |
| [2] | WG 07 MPEG 3D Graphics coding, *Common Test Conditions for V3C and V-PCC,* Online: MPEG N0038, Oct. 2020. |
| [3] | "ShapeNet," [Online]. Available: https://shapenet.org/terms. |
| [4] | "ModelNet," [Online]. Available: https://modelnet.cs.princeton.edu/. |
| [5] | "Kitti VIsion Benchmark Suite," [Online]. Available: http://www.cvlibs.net/datasets/kitti/index.php. |
| [6] | Z. W. e. al., "3d shapenets: A deep representation for volumetric shapes," *IEEE CVPR,* 2015. |
| [7] | J. Pang, M. A. Lodhi, G. Martin-Cocher and D. Tian, *[PCC] AI-based Point Cloud Compression for New PCC,* Online: MPEG m56776, Apr 2021. |
| [8] | WG 07, MPEG 3DG, *Common Test Conditions for G-PCC,* Online: MPEG N00106, Jul 2021. |
| [9] | WG 07 MPEG 3D Graphics coding, *[AI-3DGC][EE13.54-ralated] Dataset Split for AI-PCC,* Online: MPEG m58754, Jan 2022. |
| [10] | Explorations WG 07 MPEG 3D Graphics coding, *Guidelines for conducting AI exploration experiments for PCC,* Online: MPEG N00282, Jan 2022. |
| [11] | WG 07 MPEG 3D Graphics coding, *Description of Exploration Experiment 5.3 on AI-based Dynamic PC Coding,* Online: MPEG N00344, July 2022. |
| [12] | "Test-GPCC," [Online]. Available: https://github.com/yydlmzyz/Test-GPCC/. |
| [13] | "Point Cloud Utils," [Online]. Available: https://github.com/fwilliams/point-cloud-utils#estimating-normals-from-a-point-cloud. |
| [14] | M. Lodhi, J. Wang, A. Zaghetto and H. Wei, "m59830: [AI-3DGC] On Reporting Template for AI-based PCC," 27 April 2022. [Online]. Available: m59830. |
| [15] | "mpeg-pcc-ai-report," [Online]. Available: http://mpegx.int-evry.fr/software/MPEG/PCC/ai/mpeg-pcc-ai-report/. |
| [16] | A. Zaghetto, D. Graziosi and A. Tabatabai, *Discussion on Point Cloud Geometry Compression Using Machine Learning,* Online: MPEG m57301, Jul 2021. |
| [17] | WG 02, MPEG Technical requirements, *CfP for Dynamic Mesh Coding,* Online: MPEG N00145, Oct 2021. |

1. <https://mpegfs.int-evry.fr/mpegcontent/ws-mpegcontent/MPEG-I/Part05-PointCloudCompression/dataset> [↑](#footnote-ref-2)
2. Script that was used for converting point clouds from 11 bit to 10 bit: <https://github.com/ftyaaa/quantizer> [↑](#footnote-ref-3)
3. <https://github.com/CloudCompare/CloudComPy> [↑](#footnote-ref-4)
4. <https://github.com/CloudCompare/CloudCompare> [↑](#footnote-ref-5)